Module 9

Securing Administration

Contents:

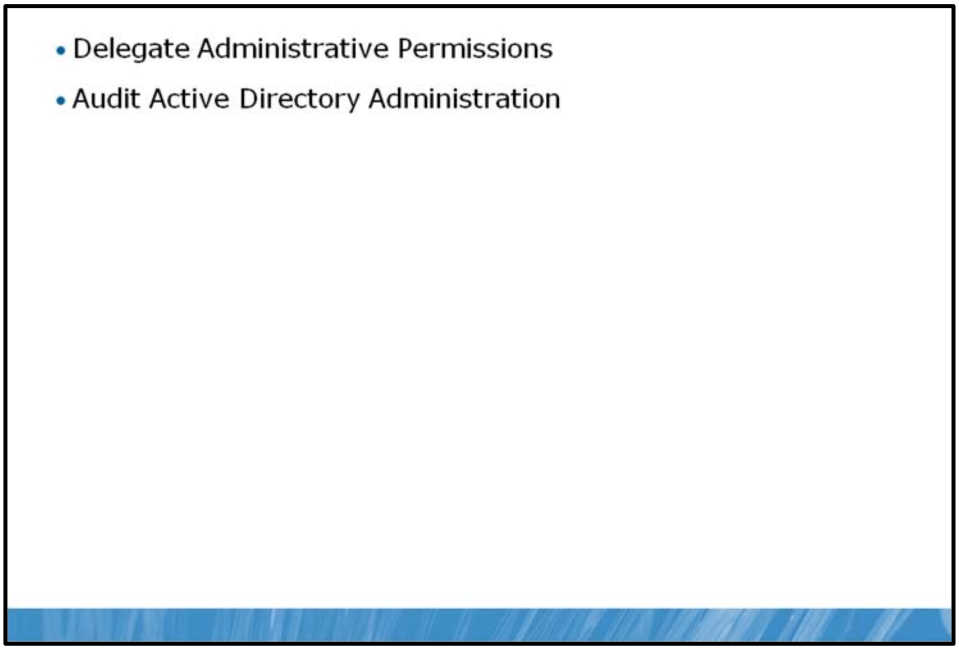
Lesson 1: Delegate Administrative Permissions **9-3**

Lab A: Delegate Administration **9-18**

Lesson 2: Audit Active Directory Administration **9-23**

Lab B: Audit Active Directory Changes **9-35**

Module Overview



- Delegate Administrative Permissions
- Audit Active Directory Administration

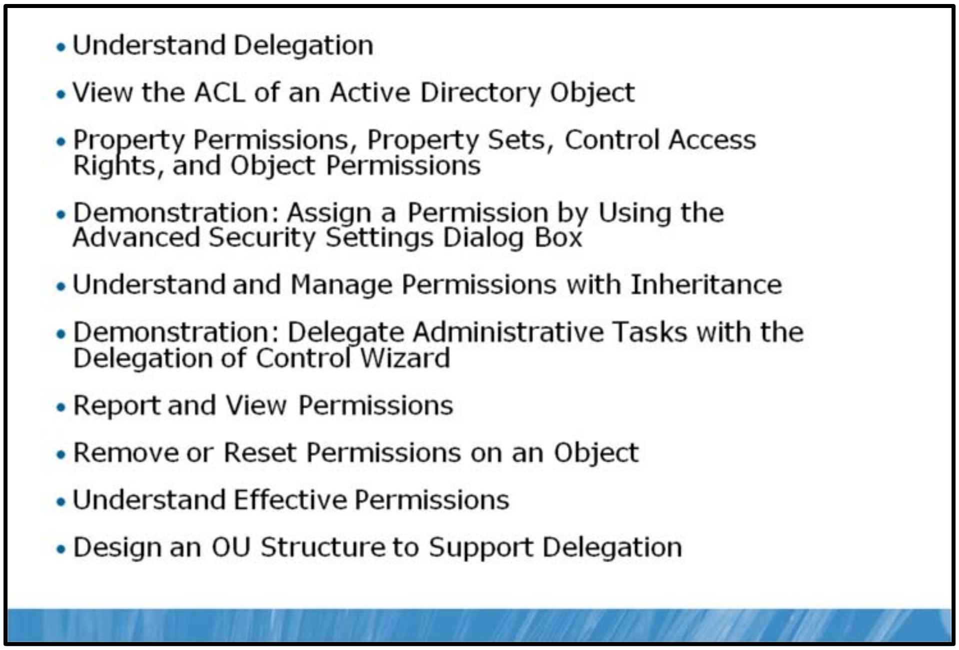
Today, security is of highest priority in most organizations. Organizations are not only removing the unnecessary administrative privileges that were assigned to users on their workstations, but are also striving to lock down and manage the privileges given to administrators themselves. To manage the security of Active Directory® administration, you need to understand how to delegate specific administrative tasks and audit changes that are made to the directory.

Objectives

After completing this module, you will be able to:

- Delegate administrative permissions.
- Audit Active Directory administration.

Lesson 1
Delegate Administrative Permissions



- Understand Delegation
- View the ACL of an Active Directory Object
- Property Permissions, Property Sets, Control Access Rights, and Object Permissions
- Demonstration: Assign a Permission by Using the Advanced Security Settings Dialog Box
- Understand and Manage Permissions with Inheritance
- Demonstration: Delegate Administrative Tasks with the Delegation of Control Wizard
- Report and View Permissions
- Remove or Reset Permissions on an Object
- Understand Effective Permissions
- Design an OU Structure to Support Delegation

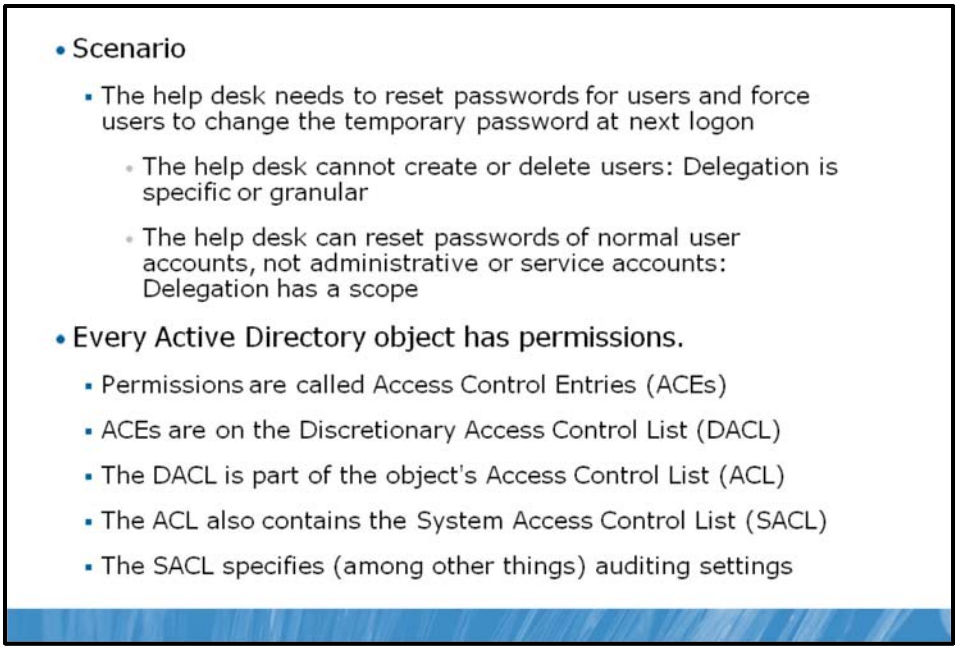
In previous modules, you learned how to create users, groups, computers, and organizational units (OUs). You also learned to access the properties of those objects. Your ability to perform those actions was dependent on your membership in the groups with administrative privileges in the domain. Every user on the help desk team need not be a member of the domain's Administrators group or other built-in groups just to reset user passwords and unlock user accounts. Instead, you can enable the help desk and each role in your organization to only perform the tasks required of the role. In this lesson, you will learn to delegate specific administrative tasks within Active Directory by changing the access control lists (ACLs) on Active Directory objects.

Objectives

After completing this lesson, you will be able to:

- Describe the business purpose of delegation.
- Assign permissions to Active Directory objects using the security editor user interfaces and the Delegation of Control Wizard.
- View and report permissions on Active Directory objects by using user-interface and command-line tools.
- Reset the permissions on an object to its default.
- Describe the relationship between delegation and OU design.

Understand Delegation



- Scenario
  - The help desk needs to reset passwords for users and force users to change the temporary password at next logon
    - The help desk cannot create or delete users: Delegation is specific or granular
    - The help desk can reset passwords of normal user accounts, not administrative or service accounts: Delegation has a scope
- Every Active Directory object has permissions.
  - Permissions are called Access Control Entries (ACEs)
  - ACEs are on the Discretionary Access Control List (DACL)
  - The DACL is part of the object's Access Control List (ACL)
  - The ACL also contains the System Access Control List (SACL)
  - The SACL specifies (among other things) auditing settings

In most organizations, there is more than one administrator, and as organizations grow, administrative tasks are often distributed among the administrators or support organizations. For example, in many organizations, the help desk can reset user passwords and unlock the user accounts that are locked out. This capability of the help desk is a delegated administrative task.

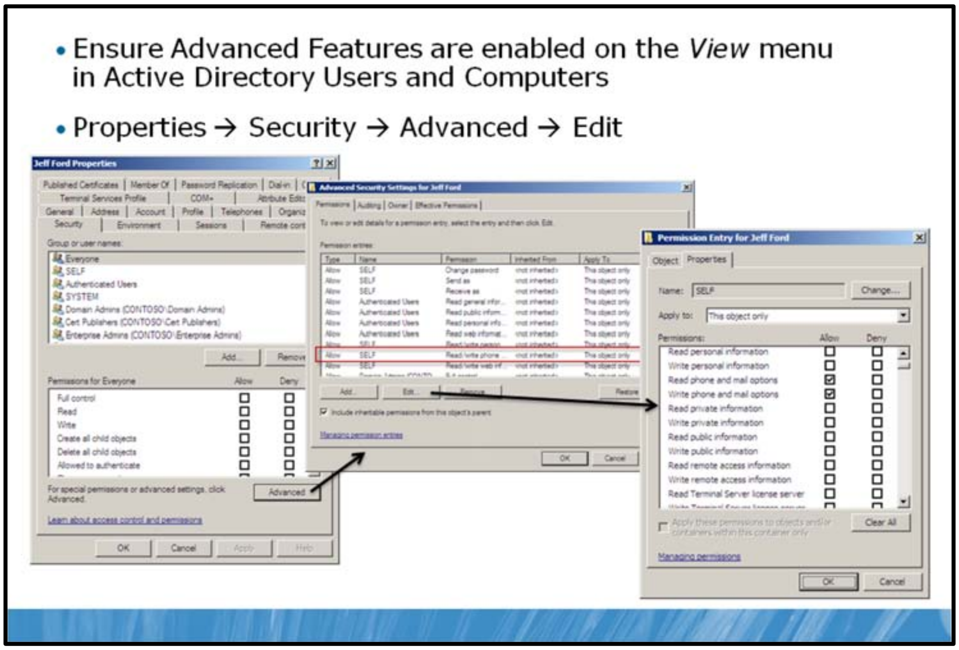
The help desk cannot usually create new user accounts, but can make specific changes to existing user accounts. The capability that is delegated is specific or granular.

In most organizations, the help desk's ability to reset passwords applies to normal user accounts, but not to accounts used for administration or service accounts. Therefore, the delegation is said to be scoped to standard user accounts.

All Active Directory objects, such as the users, computers, and groups you created in the previous module, can be secured by using a list of permissions. Therefore, you can give your help desk permission to reset passwords on user objects. The permissions on an object are called access control entries (ACEs), and they are assigned to users, groups, or computers, which are also known as security principals. ACEs are saved in the object's discretionary access control list (DACL). The DACL is a part of the object's ACL, which also contains the system access control list (SACL) that includes auditing settings.

The delegation of administrative control involves assigning permissions that manage access to objects and properties in Active Directory. Just as you can give a group the ability to change files in a folder, you can give the group the ability to reset passwords on user objects.

View the ACL of an Active Directory Object



- Ensure Advanced Features are enabled on the *View* menu in Active Directory Users and Computers
- Properties → Security → Advanced → Edit

Jeff Ford Properties

Group or user names:

- Everyone
- SELF
- Authenticated Users
- SYSTEM
- Domain Admins (CONTOSO\Domain Admins)
- Cert Publishers (CONTOSO\Cert Publishers)
- Enterprise Admins (CONTOSO\Enterprise Admins)

Permissions for Everyone	Allow	Deny
Full control	□	□
Read	□	□
Write	□	□
Create all child objects	□	□
Delete all child objects	□	□
Allowed to authenticate	□	□

For special permissions or advanced settings, click Advanced.

Advanced Security Settings for Jeff Ford

To view or edit details for a permission entry, select the entry and then click Edit.

Permission entries:

Type	Name	Permission	Inherited From	Apply To
Allow	SELF	Change password	<not inherited>	This object only
Allow	SELF	Send as	<not inherited>	This object only
Allow	SELF	Receive as	<not inherited>	This object only
Allow	Authenticated Users	Read general infor...	<not inherited>	This object only
Allow	Authenticated Users	Read public inform...	<not inherited>	This object only
Allow	Authenticated Users	Read personal info...	<not inherited>	This object only
Allow	Authenticated Users	Read web informat...	<not inherited>	This object only
Allow	SELF	Read/write person...	<not inherited>	This object only
Allow	SELF	Read/write phone ...	<not inherited>	This object only
Allow	SELF	Read/write web inf...	<not inherited>	This object only

☑ Include inheritable permissions from this object's parent

Permission Entry for Jeff Ford

Name: SELF

Apply to: This object only

Permissions:	Allow	Deny
Read personal information	□	□
Write personal information	□	□
Read phone and mail options	☑	□
Write phone and mail options	☑	□
Read private information	□	□
Write private information	□	□
Read public information	□	□
Write public information	□	□
Read remote access information	□	□
Write remote access information	□	□
Read Terminal Server license server	□	□

□ Apply these permissions to objects and/or containers within this container only

Each object in Active Directory has its own ACL. If you have sufficient permissions, you can modify the permissions to control the level of access on a specific Active Directory object. To view the ACL on an object, perform the following steps:

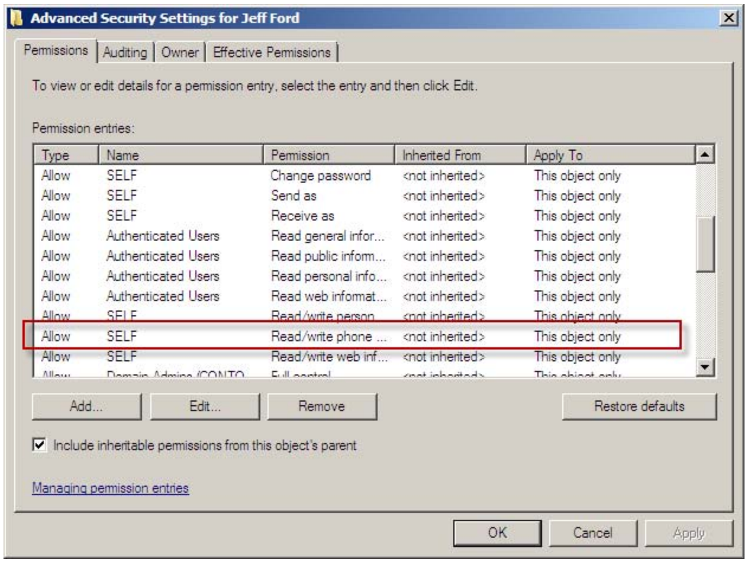
1. Open the Active Directory Users and Computers snap-in.
2. Click the **View** menu and click **Advanced Features**.
3. Right-click an object and click **Properties**.
4. Click the **Security** tab.

Note If Advanced Features is disabled, you will not see the Security tab in an object's Properties dialog box.

5. Click **Advanced**.

The Security tab shows a very high-level overview of the security principals that have been given permissions to the object. However, in the case of Active Directory ACLs, the Security tab is rarely detailed enough to provide the information you need to interpret or manage the ACL. To see a more detailed permission list, click Advanced to open the Advanced Security Settings dialog box.

The **Advanced Security Settings** dialog box appears, as shown in the following image.



Advanced Security Settings for Jeff Ford

To view or edit details for a permission entry, select the entry and then click Edit.

Permission entries:

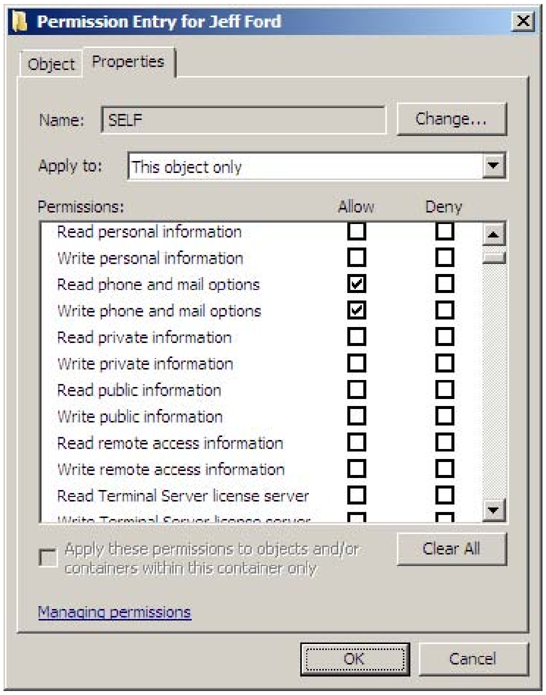
Type	Name	Permission	Inherited From	Apply To
Allow	SELF	Change password	<not inherited>	This object only
Allow	SELF	Send as	<not inherited>	This object only
Allow	SELF	Receive as	<not inherited>	This object only
Allow	Authenticated Users	Read general infor...	<not inherited>	This object only
Allow	Authenticated Users	Read public inform...	<not inherited>	This object only
Allow	Authenticated Users	Read personal info...	<not inherited>	This object only
Allow	Authenticated Users	Read web informat...	<not inherited>	This object only
Allow	SELF	Read/write person...	<not inherited>	This object only
Allow	SELF	Read/write phone ...	<not inherited>	This object only
Allow	SELF	Read/write web inf...	<not inherited>	This object only
Allow	Domain Admins (CONTO...	Full control	<not inherited>	This object only

☑ Include inheritable permissions from this object's parent

The Permissions page of the **Advanced Security Settings** dialog box shows the DACL of the object. The screen shot shows ACEs summarized on a line of the Permission entries list. In this dialog box, you do not see the granular ACEs of the DACL. For example, the permission entry that is highlighted actually consists of two ACEs.

To see the granular ACEs of a permission entry, select the entry and click **Edit**.

The **Permission Entry** dialog box appears, detailing the specific ACEs that make up the entry.



Permission Entry for Jeff Ford

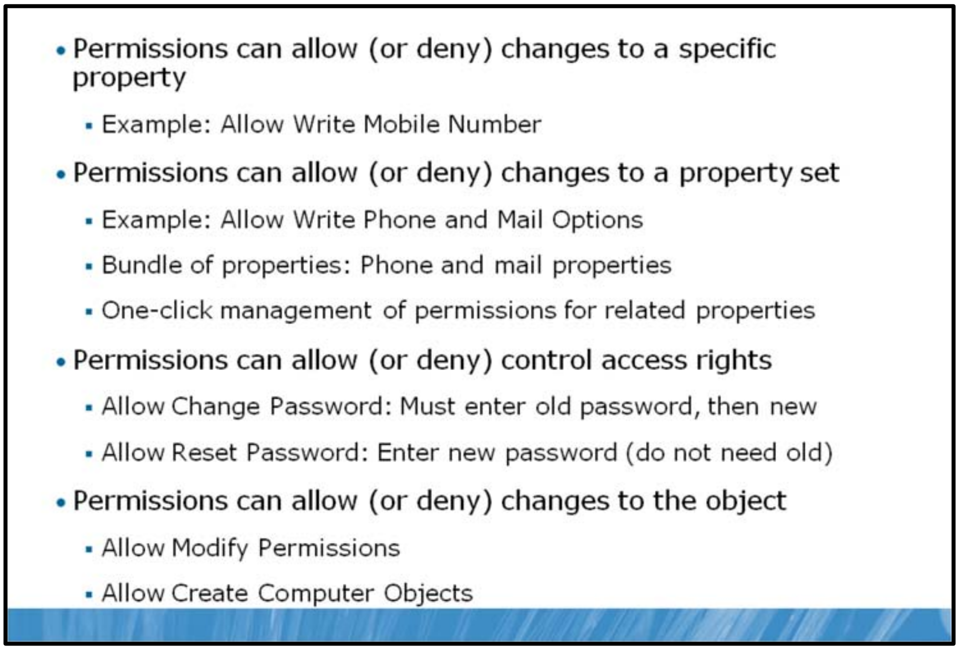
Name: SELF

Apply to: This object only

Permissions:	Allow	Deny
Read personal information	□	□
Write personal information	□	□
Read phone and mail options	☑	□
Write phone and mail options	☑	□
Read private information	□	□
Write private information	□	□
Read public information	□	□
Write public information	□	□
Read remote access information	□	□
Write remote access information	□	□
Read Terminal Server license server	□	□
Write Terminal Server license server	□	□

□ Apply these permissions to objects and/or containers within this container only

Property Permissions, Property Sets, Control Access Rights, and Object Permissions



- Permissions can allow (or deny) changes to a specific property
  - Example: Allow Write Mobile Number
- Permissions can allow (or deny) changes to a property set
  - Example: Allow Write Phone and Mail Options
  - Bundle of properties: Phone and mail properties
  - One-click management of permissions for related properties
- Permissions can allow (or deny) control access rights
  - Allow Change Password: Must enter old password, then new
  - Allow Reset Password: Enter new password (do not need old)
- Permissions can allow (or deny) changes to the object
  - Allow Modify Permissions
  - Allow Create Computer Objects

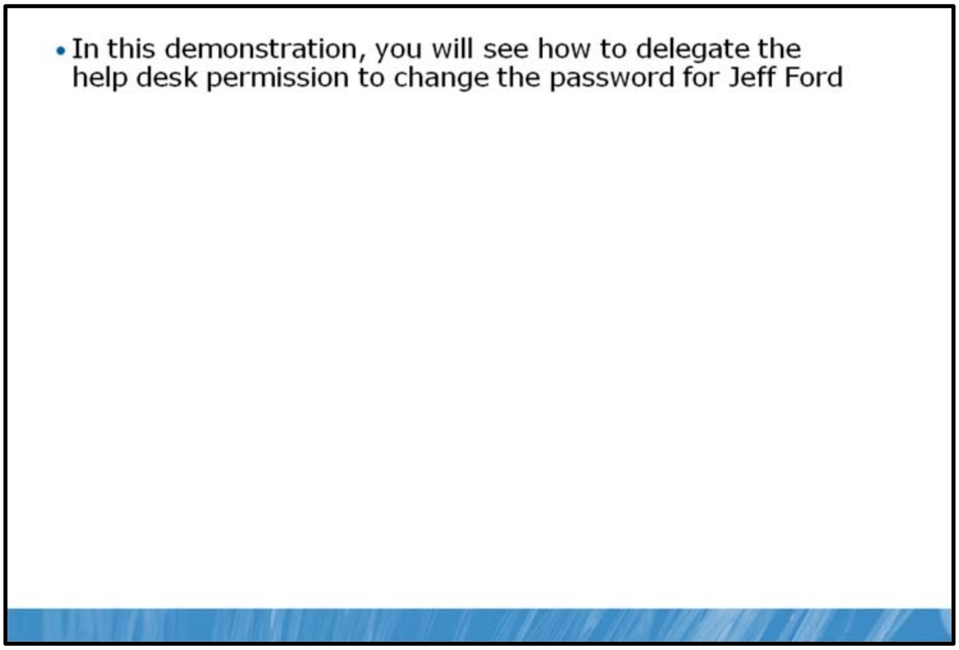
The DACL of an object allows you to assign permissions to specific properties of an object. For example, you can allow (or deny) permission to change phone and email options. This is, in fact, not just one property; it is a property set that includes multiple specific properties. Using property sets, you can easily manage permissions to commonly used collections of properties. But, you could assign more granular permissions and allow or deny permission to change just the mobile telephone number or the street address.

Permissions can also be assigned to control access rights, such as changing or resetting a password. The difference between those two control access rights is important. If you have the right to change a password, you must know and enter the current password before making the change. If you have the right to reset a password, you need not know the previous password.

Finally, permissions can be assigned to objects. For example, the ability to change permissions on an object is controlled by the Allow Modify Permissions access control entry (ACE). Object permissions also control whether you are able to create child objects. For example, you might give your desktop support team permissions to create computer objects in the Client Computers OU. The Allow Create Computer Objects ACE would be assigned to the desktop support team at the OU.

You can manage the type and scope of permissions by using the Object tab and the Properties tab, and the Apply To drop-down lists on each tab.

Demonstration: Assign a Permission by Using the Advanced Security Settings Dialog Box



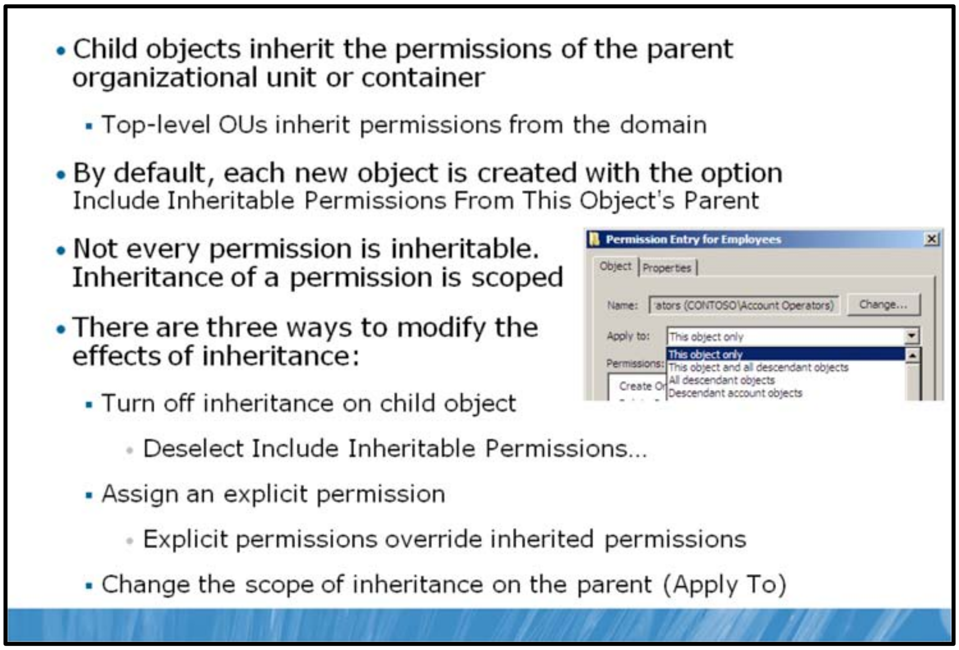
- In this demonstration, you will see how to delegate the help desk permission to change the password for Jeff Ford

Consider that you want to allow the help desk to change the password only on Jeff Ford's user account. In this section, you will learn to do it in the most complicated way first. You will assign the ACE on the DACL of the user object. Then, you will learn to delegate by using the Delegation of Control Wizard for the entire OU of users. Finally, you will see why this latter practice is recommended.

Demonstration Steps

- Enable **Advanced View** in the Active Directory Users and Computers console.
- Open **Advanced Security Properties** of the user account object.
- Delegate permission to reset the password.

Understand and Manage Permissions with Inheritance



- Child objects inherit the permissions of the parent organizational unit or container
  - Top-level OUs inherit permissions from the domain
- By default, each new object is created with the option Include Inheritable Permissions From This Object's Parent
- Not every permission is inheritable. Inheritance of a permission is scoped
- There are three ways to modify the effects of inheritance:
  - Turn off inheritance on child object
    - Deselect Include Inheritable Permissions...
  - Assign an explicit permission
    - Explicit permissions override inherited permissions
  - Change the scope of inheritance on the parent (Apply To)

Permission Entry for Employees

Name: ators (CONTOSO\Account Operators)

Apply to: This object only

This object only
This object and all descendant objects
All descendant objects
Descendant account objects

Assigning the help desk permission to reset passwords for each individual user object is tedious. But, in Active Directory, it is not a good practice to assign permissions to individual objects. Instead, you should assign permissions at the level of organizational units. The permissions you assign to an OU will be inherited by all objects in the OU. Therefore, if you give the help desk permission to reset passwords for user objects and attach that permission to the OU that contains the users, all user objects within that OU will inherit that permission. In just one step, you can delegate that administrative task.

Child objects inherit the permissions of the parent container or OU. That container or OU in turn inherits its permissions from its parent container OU. If it is a first-level container or OU, it inherits the permissions from the domain itself. The reason child objects inherit permissions from their parents is that, by default, each new object is created with the Include inheritable permissions from this object's parent option enabled.

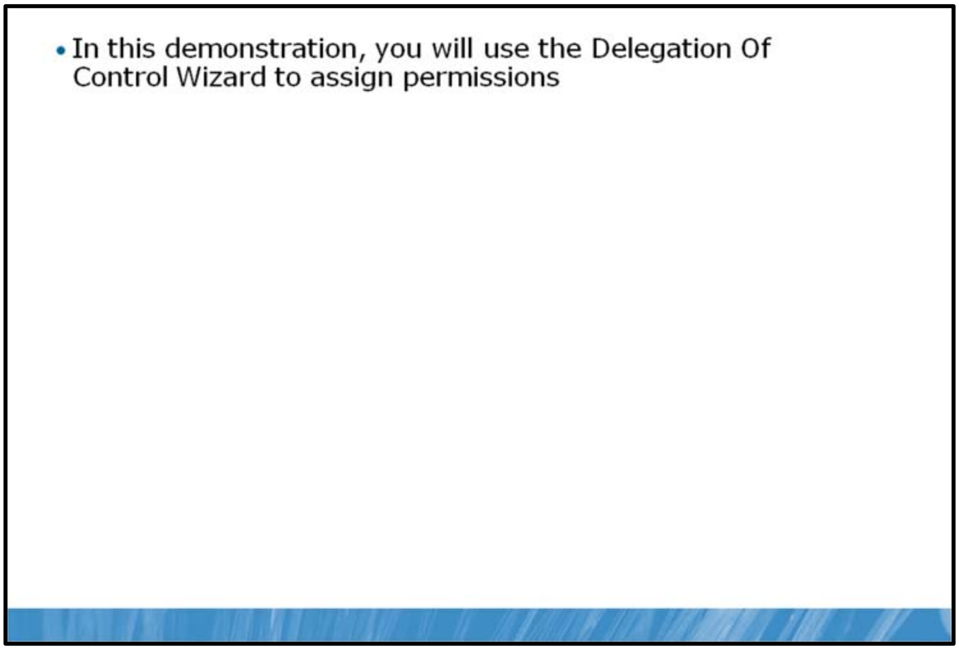
However, note that as the option indicates, only inheritable permissions will be inherited by the child object. Not all permissions are inheritable. For example, the permission to reset passwords, when assigned to an OU, would not be inherited by group objects because group objects do not have a password attribute. So, inheritance can be scoped to specific object classes: passwords are applicable to user objects, not groups. Additionally, you can use the Apply To box of the Permission Entry dialog box to scope the inheritance of a permission. The conversation can start to get very complicated. What you should know is that, by default, new objects inherit inheritable permissions from their parent object—usually, an OU or a container.

What if the permission that is being inherited is not appropriate? You can do the following three things to modify the permissions that a child object is inheriting:

- First, you can disable inheritance by deselecting the Include Inheritable Permissions From This Object's Parent option in the Advanced Security Settings dialog box. When you do, the object will no longer inherit any permissions from its parent; all permissions will be explicitly defined for the child object. This is generally not a good practice, because it creates an exception to the rule that is created by permissions of parent containers.

- The second option is to allow inheritance, but to override the inherited permission with a permission assigned specifically to the child object—an explicit permission. Explicit permissions always override permissions that are inherited from parent objects. This has an important implication: an explicit permission that allows access will actually override an inherited permission that denies the same access. The rule (Deny) is being defined by a parent, but the child object has been configured to be an "exception" (Allow).
- Finally, you can change the scope of inheritance on the parent permission itself by changing the option in the **Apply To** drop-down list in the **Permission Entry** dialog box. In most cases, this is the best practice. What you are doing, in effect, is defining the security policy in the form of the ACL more accurately at its source, rather than trying to override it further down the tree.

Demonstration: Delegate Administrative Tasks with the Delegation of Control Wizard



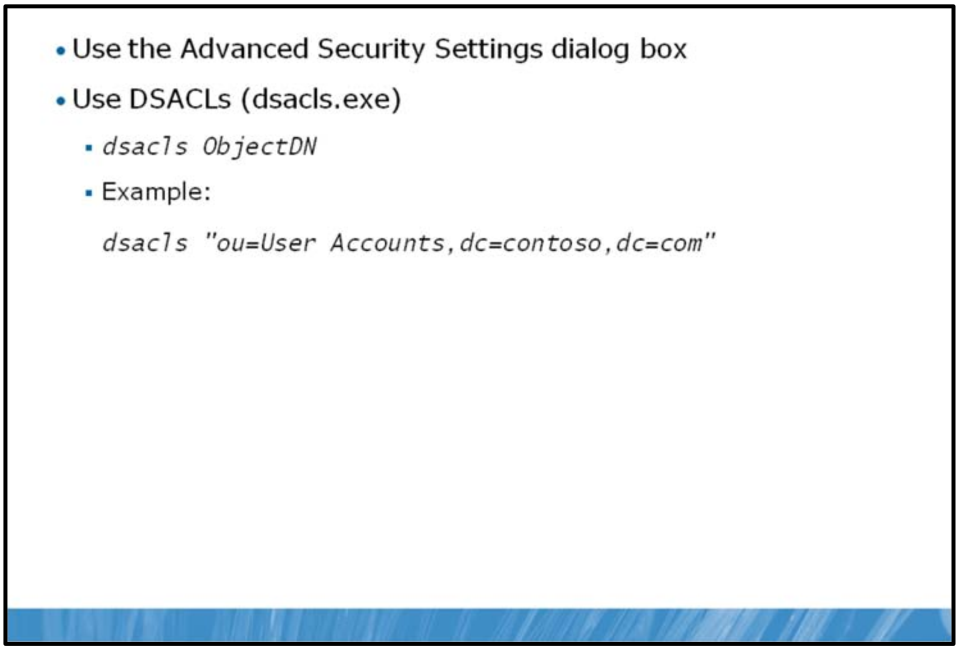
- In this demonstration, you will use the Delegation Of Control Wizard to assign permissions

You have seen the complexity of the DACL and understood that managing permissions by using the **Permission Entry** dialog box is not a simple task. Luckily, the best practice is not to manage permissions by using security interfaces, but using the Delegation Of Control Wizard. This wizard allows you to delegate several permissions on the OU level, without editing the DACL directly, but by answering questions in a wizard. However, the result is the same. After the wizard completes, it initiates a script that edits the DACL of the OU. The following procedure details the use of the wizard.

Demonstration Steps

- Run Delegation of Control Wizard on an OU.
- Delegate permissions to reset the password and force a password change on the OU level.

Report and View Permissions



- Use the Advanced Security Settings dialog box
- Use DSACLs (dsacls.exe)
  - *dsacls ObjectDN*
  - Example:

    dsacls "ou=User Accounts,dc=contoso,dc=com"

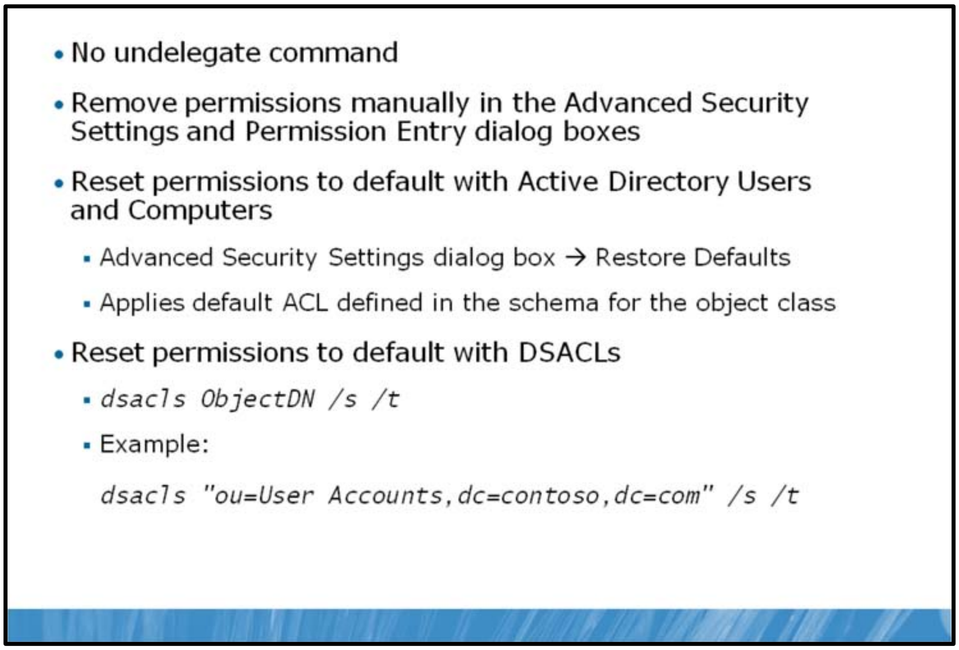
There are several other ways to view and report permissions when you need to know who can do what. You have already seen that you can view permissions on the DACL by using the Advanced Security Settings and Permission Entry dialog boxes.

DSACLs (dsacls.exe) is also available as a command-line tool that reports on directory service objects. If you type the command followed by the distinguished name of an object you will see a report of the object's permissions. For example, the following command produces a report of the permissions associated with the User Accounts OU:

```
dsacls.exe "ou=User Accounts,dc=contoso,dc=com"
```

DSACLs can also be used to set permissions—to delegate. Type dsacls.exe /? for help regarding the syntax and utilization of DSACLs.

Remove or Reset Permissions on an Object



- No undelegate command
- Remove permissions manually in the Advanced Security Settings and Permission Entry dialog boxes
- Reset permissions to default with Active Directory Users and Computers
  - Advanced Security Settings dialog box → Restore Defaults
  - Applies default ACL defined in the schema for the object class
- Reset permissions to default with DSACLs
  - *dsacls ObjectDN /s /t*
  - Example:

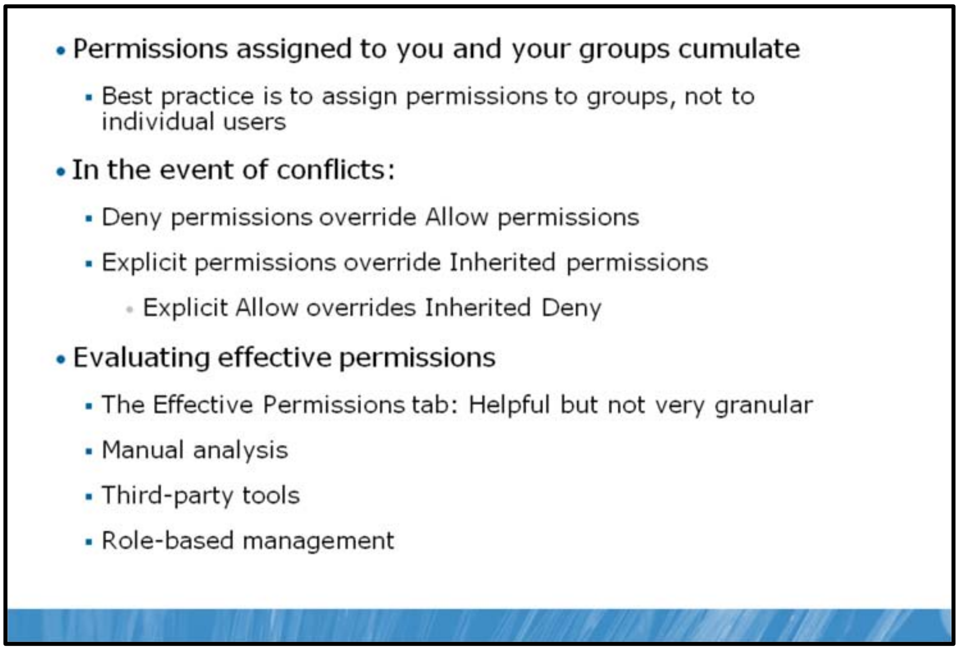
    dsacls "ou=User Accounts,dc=contoso,dc=com" /s /t

How do you remove or reset permissions that have been delegated? Unfortunately, there is no undelegate command. You must do one of the following:

- Open the Advanced Security Settings and Permission Entry dialog boxes to remove permissions.
- If you want to reset the permissions on the object back to the defaults, open the Advanced Security Settings dialog box and click Restore Defaults**.** The default permissions are defined by the Active Directory schema for the class of object. After restoring the defaults, you can reconfigure the explicit permissions you want to add to the DACL.
- DSACLs also provides the /s switch to reset permissions to the schema-defined defaults, and the /t switch to make the change for the entire “tree”—the object and all of its child objects. For example, to reset permissions on the People OU and all of its child OUs and objects, you would enter:

```
dsacls "ou=User Accounts,dc=contoso,dc=com" /s /t
```

Understand Effective Permissions



- Permissions assigned to you and your groups cumulate
  - Best practice is to assign permissions to groups, not to individual users
- In the event of conflicts:
  - Deny permissions override Allow permissions
  - Explicit permissions override Inherited permissions
    - Explicit Allow overrides Inherited Deny
- Evaluating effective permissions
  - The Effective Permissions tab: Helpful but not very granular
  - Manual analysis
  - Third-party tools
  - Role-based management

Effective permissions are the resulting permissions for a security principal, such as a user or group, based on the cumulative effect of each inherited and explicit ACE. Your ability to reset a user's password, for example, may be due to your membership in a group that is allowed the Reset Password permission on an OU several levels above the user object. The inherited permission assigned to a group to which you belong results in an effective permission of Allow: Reset Password. Your effective permissions can be complicated when you consider Allow and Deny permissions, explicit and inherited ACEs, and the fact that you may belong to multiple groups, each of which may be assigned different permissions.

To calculate effective permissions for a specific user or a group, an Active Directory object, or for a file or folder, you can follow the following simple procedure:

1. Right-click the object, file or folder, click **Properties**, and then click the **Security** tab.
2. Click **Advanced**, click the **Effective Permissions** tab, and then click **Select**.
3. In **Enter the object name to select**, enter the name of a user or group, and then click **OK**. The selected check boxes indicate the effective permissions of the user or group for that file or folder.

Permissions, whether assigned to your user account or a group to which you belong, are equivalent. In the end, an ACE applies to you, the user. The best practice is to manage permissions by assigning them to groups, but it is also possible to assign ACEs to individual users or computers. A permission that has been assigned directly to you, the user, is neither more important nor less important than a permission assigned to a group to which you belong.

Allow permissions, which allow access, are cumulative. When you belong to several groups, and those groups have been granted permissions that allow a variety of tasks, you will be able to perform all of the tasks assigned to all of those groups, as well as tasks assigned directly to your user account.

Deny permissions, which deny access (), override equivalent Allow permissions. If you are in one group that has been allowed the permission to reset passwords, and another group that has been denied permission to reset passwords, the Deny permission prevents you from resetting passwords.

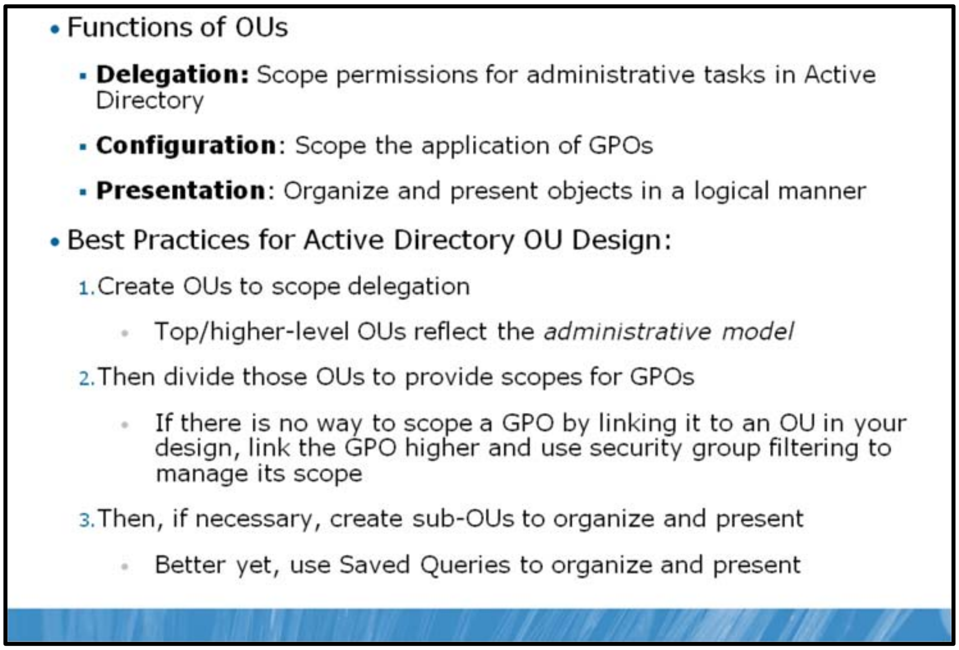
Note It is unnecessary to assign Deny permissions. If you do not assign an Allow permission, users cannot perform the task. Before assigning a Deny permission, check to see if you could achieve your goal by removing an Allow permission instead. Use Deny permissions rarely. For example, if you want to delegate an Allow permission to a group, but exempt only one member from that group, you can use a Deny permission on that specific user account while the group will still have Allow permission.

Each permission is granular. Even if you have been denied the ability to reset passwords, you may still have the ability, through other Allow permissions, to change the user's logon name or email address.

In this lesson, you learned that child objects inherit the inheritable permissions of parent objects by default, and that explicit permissions can override inheritable permissions. This means that an explicit Allow permission will actually override an inherited Deny permission.

Unfortunately, the complex interaction of user, group, explicit, inherited, Allow, and Deny permissions can make evaluating effective permissions tedious. You can use the permissions reported by the DSACLs command or on the **Permissions** tab of the **Advanced Security Settings** dialog box to begin evaluating effective permissions, but it will be a manual task.

Design an OU Structure to Support Delegation



- Functions of OUs
  - **Delegation:** Scope permissions for administrative tasks in Active Directory
  - **Configuration**: Scope the application of GPOs
  - **Presentation**: Organize and present objects in a logical manner
- Best Practices for Active Directory OU Design:
  1. Create OUs to scope delegation
     - Top/higher-level OUs reflect the *administrative model*
  2. Then divide those OUs to provide scopes for GPOs
     - If there is no way to scope a GPO by linking it to an OU in your design, link the GPO higher and use security group filtering to manage its scope
  3. Then, if necessary, create sub-OUs to organize and present
     - Better yet, use Saved Queries to organize and present

OUs are, as you now know, administrative containers. They contain objects that share similar requirements for administration, configuration, and visibility. You now understand the first of those requirements: administration. Objects that administrators administer should be contained within a single OU. By placing your users in a single OU perhaps called User Accounts, you could delegate the help desk permission to change all users' passwords by assigning one permission to one OU. Any other permissions that affect what an administrator can do to a user object would be assigned in the User Accounts OU. For example, you might allow your Human Resources managers to disable user accounts in the event of an employee's termination. You would delegate that permission, again, to the User Accounts OU.

Remember that administrators should be logging on to their systems with user credentials and launching administrative tools with the credentials of a secondary account that has appropriate permissions to perform administrative tasks. Secondary accounts are the administrative accounts of the enterprise. It is not appropriate for the front-line help desk to be able to reset passwords on such privileged accounts, and you probably would not want Human Resources managers to disable administrative accounts. Therefore, administrative accounts should be administered differently than "normal" user accounts. That's why you would have a separate OU, such as Admins, for administrative user objects, which would be delegated quite differently than the User Accounts OU.

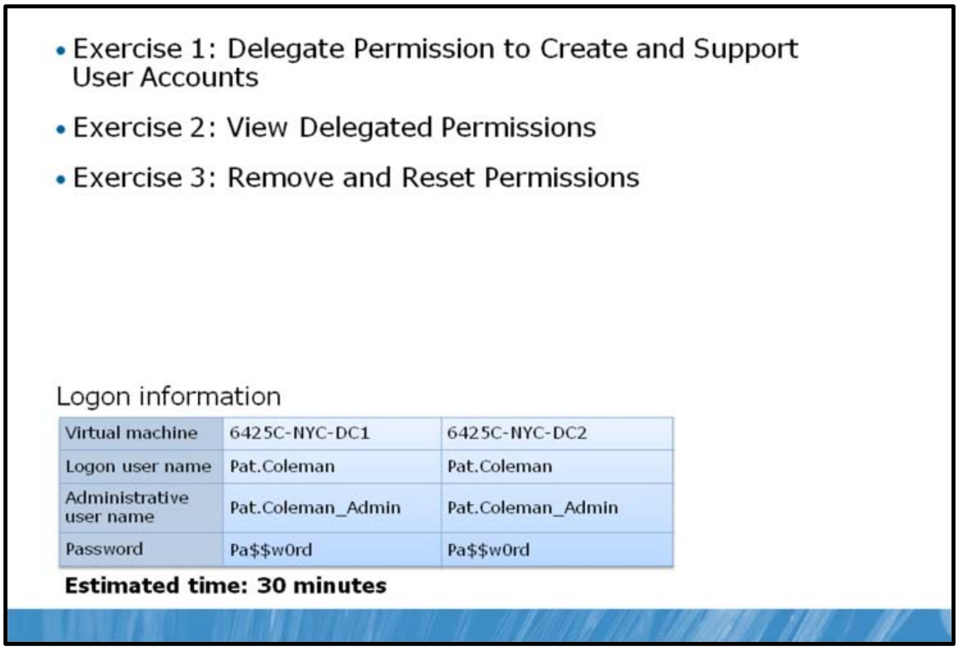
Similarly, you might delegate to the desktop support team the ability to add computer objects to an OU called Client Computers, which contains your desktops and laptops, but not to the Servers OU, where only the Server Administration group has permissions to create and manage computer objects.

The primary role of OUs is to scope delegation—to apply permissions to objects and sub-OUs. When you design an Active Directory environment, you always begin by designing an OU structure that makes delegation efficient—a structure that reflects the administrative model of your organization. Rarely does object administration in Active Directory look like your organizational chart. Typically, all normal user accounts are supported the same way, by the same team—so, user objects are often found in a single OU or a single OU branch. Quite often, an organization that has a centralized help desk function to support users will also have a centralized desktop support function. In this case, all client computer objects would be within a single OU or a single OU branch. But, if desktop support is decentralized, it would be likely the Client Computers OU are divided into sub-OUs, representing geographic locations. Each location would be delegated to allow the local support team to add computer objects to the domain in that location.

Design OUs first to enable the efficient delegation of objects in the directory. After you have achieved that design, you can refine the design to facilitate the configuration of computers and users through Group Policy.

Also, you can consider placing access permissions groups within separate OUs. As a best practice, access permissions groups should be placed in OUs that deny read permissions to standard users so that these groups do not appear in search results when standard users search the directory. Using this approach, you can make these groups visible only to administrators and people who can manage their group membership.

Lab A: Delegate Administration



- Exercise 1: Delegate Permission to Create and Support User Accounts
- Exercise 2: View Delegated Permissions
- Exercise 3: Remove and Reset Permissions

Logon information

Virtual machine	6425C-NYC-DC1	6425C-NYC-DC2
Logon user name	Pat.Coleman	Pat.Coleman
Administrative user name	Pat.Coleman_Admin	Pat.Coleman_Admin
Password	Pa$$w0rd	Pa$$w0rd

Estimated time: 30 minutes

Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. On the host computer, click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In Hyper-V™ Manager, click **6425C-NYC-DC1**, and in the Actions pane, click **Start**.
3. In the Actions pane, click **Connect**. Wait until the virtual machine starts.
4. Log on by using the following credentials:
   - User name: **Pat.Coleman**
   - Password: **Pa$$w0rd**
   - Domain: **Contoso**
5. Open Windows Explorer and then browse to **D:\Labfiles\Lab09a**.
6. Run **Lab09a_Setup.bat** with administrative credentials. Use the account **Pat.Coleman_Admin** with the password **Pa$$w0rd**.
7. The lab setup script runs. When it is complete, press any key to continue.
8. Close the Windows Explorer window, **Lab09a**.
9. Repeat steps 2-4 for **6425C-NYC-DC2**.

Lab Scenario

The enterprise security team at Contoso, Ltd has asked you to lock down the administrative permissions delegated to support personnel.

Exercise 1: Delegate Permission to Create and Support User Accounts

In this exercise, you will delegate permission to the help desk to unlock user accounts, reset passwords, and force users to change passwords at the next logon. This permission will scope only to standard user accounts and will not allow the help desk to change passwords of administrative accounts. You will also delegate permission to the User Account Admins group to create and delete user accounts, as well as full control over user accounts.

The main tasks for this exercise are as follows:

1. Create security groups for role-based management.
2. Delegate control of user support with the Delegation of Control Wizard.
3. Delegate permission to create and delete users with the Access Control List Editor interface.
4. Validate the implementation of delegation.

▶ Task 1: Create security groups for role-based management.

1. On NYC-DC2, run **Active Directory Users and Computers** with administrative credentials. Use the account **Pat.Coleman_Admin** with the password **Pa$$w0rd**.
2. In the **Groups\Role** OU, create the following role groups:
   - **Help Desk** (global security group)
   - **User Account Admins** (global security group)
3. Add the following users' administrative accounts to the **Help Desk** group. Be careful not to add the users' standard, non-privileged account.
   - **Aaron M. Painter**
   - **Elly Nkya**
   - **Julian Price**
   - **Holly Dickson**
4. Add the following users' administrative accounts to the **User Account Admins** group. Be careful not to add the users' standard, non-privileged account.
   - **Pat Coleman**
   - **April Meyer**
   - **Max Stevens**

▶ Task 2: Delegate control of user support with the Delegation Of Control Wizard.

- Right-click the **User Accounts** OU and then click **Delegate Control**. Delegate to the **Help Desk** group the permission to reset user passwords and force users to change passwords at next logon.

▶ Task 3: Delegate permission to create and delete users with the Access Control List Editor interface.

1. Turn on the **Advanced Features** view of the **Active Directory Users and Computers** snap-in.
2. Right-click the **User Accounts** OU, and then click **Properties**. Click the **Security** tab, and then click **Advanced**.
3. Add permissions that give **User Account Admins** the ability to create and delete users and full control over user objects. Be careful to limit the **Full Control** permission to descendant user objects only.

▶ Task 4: Validate the implementation of delegation.

1. Close Active Directory Users and Computers.
2. Run **Active Directory Users and Computers** as an administrator, with the username **Aaron.Painter_Admin** and the password **Pa$$w0rd**.
3. Confirm that you can reset the password for **Jeff Ford**, in the **Employees** OU, and that you can force him to change his password at the next logon.
4. Confirm that you cannot disable Jeff Ford's account.
5. Confirm that you cannot reset the password for **Pat Coleman (Admin)** in the **Admin Identities** OU.
6. Close Active Directory Users and Computers.
7. Run **Active Directory Users and Computers** as an administrator, with the user name **April.Meyer_Admin** and the password **Pa$$w0rd**.
8. Confirm that you can create a user account in the **Employees** OU by creating an account with your own first and last name, the user name First. Last, and the password **Pa$$w0rd**.
9. Close Active Directory Users and Computers.

Results: In this exercise, you delegated to the help desk the permission to unlock user accounts, reset passwords, and force users to change passwords at next logon through the help desk's membership in the Help Desk group. You have also delegated full control of user objects to User Account Admins group. And, you tested both delegations to validate their functionality.

Exercise 2: View Delegated Permissions

In this exercise you will view, report, and evaluate the permissions that have been assigned to Active Directory objects.

The main tasks for this exercise are as follows:

1. View permissions in the Access Control List Editor interfaces.
2. Report permissions by using DSACLs.
3. Evaluate effective permissions.

▶ Task 1: View permissions in the Access Control List Editor interfaces.

1. On NYC-DC2, run **Active Directory Users and Computers** as an administrator, with the user name **Pat.Coleman_Admin** and the password **Pa$$w0rd**.
2. Right-click the **User Accounts** OU, and then click **Properties**. Click the **Security** tab, and then click **Advanced.**
3. Sort so that permissions are displayed according to the group to which they are assigned.

   Question: How many permission entries were created for the Help Desk group by the Delegation Of Control Wizard? Is it easy to tell what permissions were assigned in the Permission Entries list? List the permissions assigned to Help Desk.

▶ Task 2: Report permissions by using DSACLs.

From the command prompt, use DSACLs to report the permissions assigned to the **User Accounts** OU. Type the following command, and then press **Enter**.

```
dsacls "ou=User Accounts,dc=contoso,dc=com"
```

Question: Which permissions are reported for Help Desk by the DSACLs command?

▶ Task 3: Evaluate effective permissions.

1. Right-click the **User Accounts** OU, and then click **Properties**. Click the **Security** tab, and then click **Advanced**.
2. Using the **Advanced Security Settings** dialog box, evaluate the **Effective Permissions** for **April.Meyer_Admin**. Locate the permissions that allow the user to create and delete users.

   Question: Do you see the Reset Password in this list?

3. In the **Employees** OU, right-click the user account for **Aaron Lee**, and then click **Properties.** Click the **Security** tab, and then click **Advanced**.
4. Using the **Advanced Security Settings** dialog box, evaluate the **Effective Permissions** for **Aaron.Painter_Admin**. Locate the permissions that allow the user to reset the password for **Aaron Lee.**

Results: In this exercise, you confirmed that the permissions you assigned in the previous exercise were applied successfully.

Exercise 3: Remove and Reset Permissions

In this exercise, you will remove delegated permissions and will reset an OU to its schema-defined default ACL.

The main tasks for this exercise are as follows:

1. Remove permissions assigned to Help Desk.
2. Reset the User Accounts OU to its default permissions.

▶ Task 1: Remove permissions assigned to Help Desk.

1. Right-click the **User Accounts** OU, and then click **Properties**. Click the **Security** tab, and then click **Advanced**.
2. Sort so that permissions are displayed according to the group to which they are assigned.
3. Remove the permissions assigned to **Help Desk.**

▶ Task 2: Reset the User Accounts OU to its default permissions.

1. Right-click the **User Accounts** OU, and then click **Properties**. Click the **Security** tab, and then click **Advanced**.
2. Click **Restore defaults**, and then click **Apply**.

   Question: What do you achieve by clicking **Reset To Default**? What permissions remain?

Results: In this exercise, you have reset the permissions on the User Accounts OU to its schema-defined defaults.

Note Do not shut down the virtual machine after you finish this lab, because the settings you have configured here will be used in the subsequent lab.

Lab Review Questions

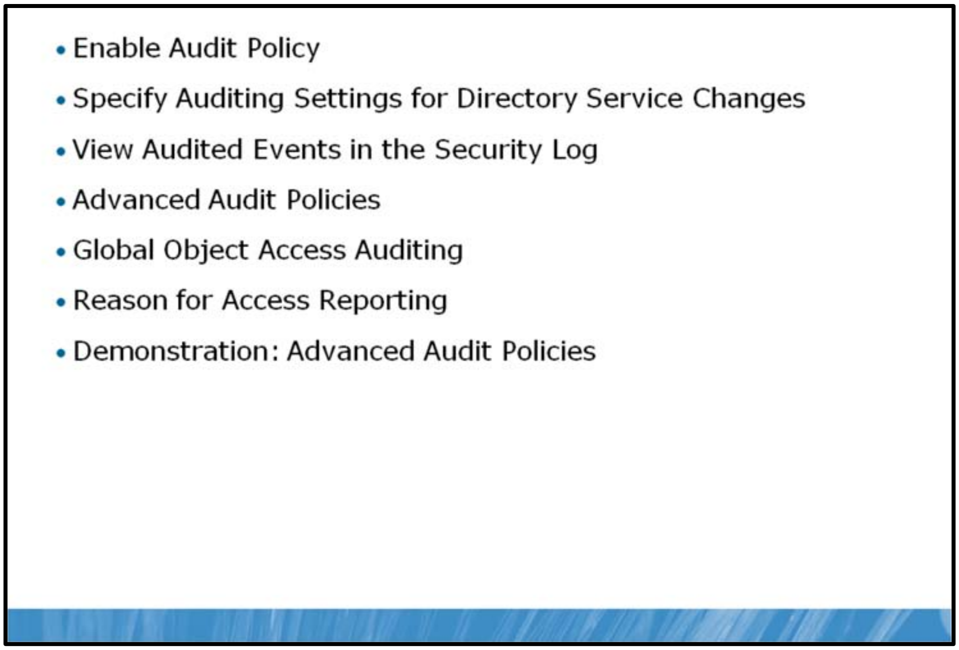
Question: When you evaluated the effective permissions for April Meyer on the User Accounts OU, why didn't you see permissions such as Reset Password in this list? Why did the permission appear when you evaluated effective permissions for Aaron Painter on Aaron Lee's user account?

Question: Does Windows make it easy to answer the following questions:

- Who can reset user passwords?
- What can XXX do as an administrator?

Question: What is the impact of resetting the ACL of an OU back to its schema-defined default?

Lesson 2
Audit Active Directory Administration



- Enable Audit Policy
- Specify Auditing Settings for Directory Service Changes
- View Audited Events in the Security Log
- Advanced Audit Policies
- Global Object Access Auditing
- Reason for Access Reporting
- Demonstration: Advanced Audit Policies

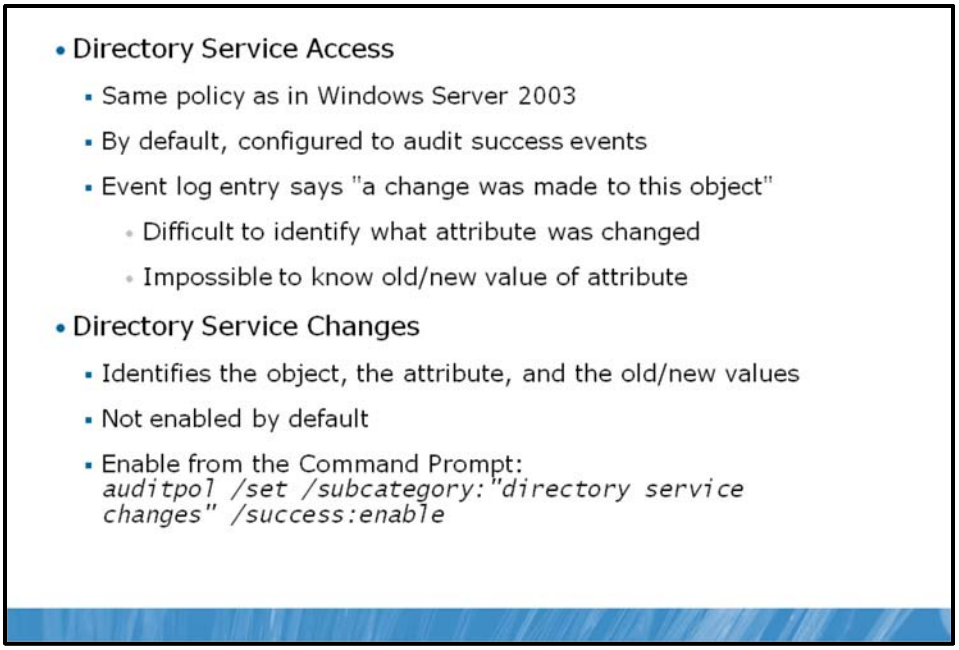
Just as auditing file and folder access allows you to log attempts to access those types of objects, the Audit Directory Service Access policy allows you to log attempts to access objects in Active Directory. Windows Server® 2008 introduces another class of auditing for Active Directory: Directory Service Changes. In addition, there are several auditing enhancements in Windows Server2008 R2 and Windows7 that increase the level of detail in security auditing logs and simplify the deployment and management of auditing policies.

Objectives

After completing this lesson, you will be able to:

- Configure audit policy to enable Directory Service Changes auditing.
- Specify auditing settings on Active Directory objects.
- Identify event log entries created by Directory Access auditing and Directory Service Changes auditing.
- Describe Advanced Audit Policies.
- Describe Global Object Access auditing.
- Describe the reason for Access Reporting.

Enable Audit Policy



- Directory Service Access
  - Same policy as in Windows Server 2003
  - By default, configured to audit success events
  - Event log entry says "a change was made to this object"
    - Difficult to identify what attribute was changed
    - Impossible to know old/new value of attribute
- Directory Service Changes
  - Identifies the object, the attribute, and the old/new values
  - Not enabled by default
  - Enable from the Command Prompt:
    auditpol /set /subcategory:"directory service changes" /success:enable

Just as the Audit Object Access policy allows you to log attempts to access objects, such as files and folders, the Audit Directory Service Access policy allows you to log attempts to access objects in Active Directory. The same basic principles apply. You configure the policy to audit Success or Failure followed by configuring the SACL of the Active Directory object to specify the types of access you want to audit.

As an example, if you want to monitor changes to the membership of a security-sensitive group, such as Domain Admins, you can enable the Audit Directory Service Access policy to audit Success events. Then, you can open the SACL of the Domain Admins group and configure an auditing entry for successful modifications of the group's Members attribute. In fact, in Windows Server 2008, the default configuration is to audit Success events for Directory Service Access and audit all changes to the Domain Admins group!

In Windows Server 2003 and Windows 2000 Server, you could audit directory service access, and you would be notified that an object, or the property of an object, had been changed, but you could not identify the previous and new values of the attribute that had changed. For example, an event could be logged indicating that a particular user changed an attribute of Domain Admins, but you could not easily identify which attribute was changed, and there was no way to determine from the audit log exactly what change was made to that attribute.

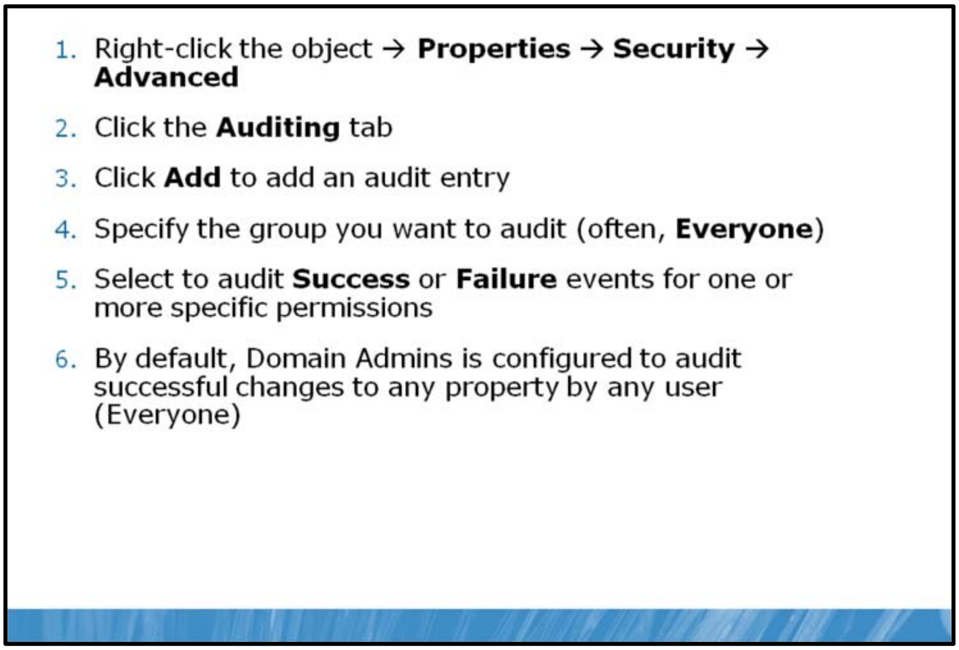
Windows Server 2008 adds an auditing category called Directory Service Changes. The important distinction between Directory Service Changes and Directory Service Access is that with Directory Service Changes auditing, you can identify the previous and current values of a changed attribute.

Directory Service Changes is not enabled in Windows Server 2008by default. Instead, Directory Service Access is enabled to mimic the auditing functionality of previous versions of Windows. To enable auditing of successful Directory Service Changes, open a command prompt on a domain controller and enter this command.

```
auditpol /set /subcategory:"directory service changes" /success:enable
```

Although you can use the preceding command to enable Directory Service Changes auditing in a lab and explore the events that are generated, we recommend that you don’t implement this in a domain until you evaluate this feature in test environment.

Specify Auditing Settings for Directory Service Changes



1. Right-click the object → **Properties** → **Security** → **Advanced**
2. Click the **Auditing** tab
3. Click **Add** to add an audit entry
4. Specify the group you want to audit (often, **Everyone**)
5. Select to audit **Success** or **Failure** events for one or more specific permissions
6. By default, Domain Admins is configured to audit successful changes to any property by any user (Everyone)

You must still modify the SACL of objects to specify which attributes should be audited.

To access the SACL and its audit entries, perform the following steps:

1. Open the **Properties** dialog box of the object you wish to audit.
2. Click the **Security** tab.
3. Click the **Advanced** button.
4. Click the **Auditing** tab.

To add an audit entry, perform the following steps:

1. Click the **Add** button.
2. Select the user, group, or computer to audit. Often, this will be the **Everyone** group.
3. In the **Auditing Entry** dialog box, indicate the type of access to audit.

   You can audit for successes, failures, or both as the specified user, group, or computer attempts to access the resource by using one or more of the granular access levels.

You can audit **Successes** to perform the following tasks:

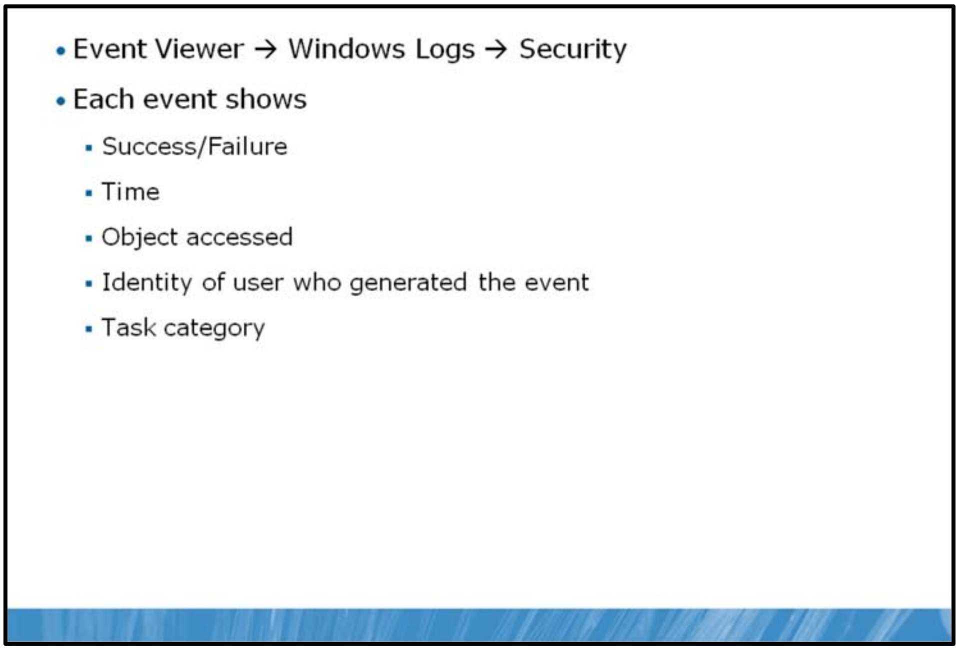
- Log resource access for reporting and billing
- Monitor access that would suggest users are performing actions greater than what you had planned, indicating that permissions are too generous
- To identify access that is out of character for a particular account, which might be a sign that a user account has been breached by a hacker

Auditing failed events allows you to:

- Monitor for malicious attempts to access resources to which access has been denied.
- Identify failed attempts to access a file or a folder to which a user does require access. This would indicate that the permissions are not sufficient to achieve a business requirement.

Note Audit logs have the tendency to get large quite rapidly, so a golden rule for auditing is to configure the bare minimum required to achieve the task. Specifying to audit the successes and failures on an active data folder for the Everyone group by using Full Control (all permissions) generates enormous audit logs that could affect the performance of the server and make locating a specific audited event impossible.

View Audited Events in the Security Log

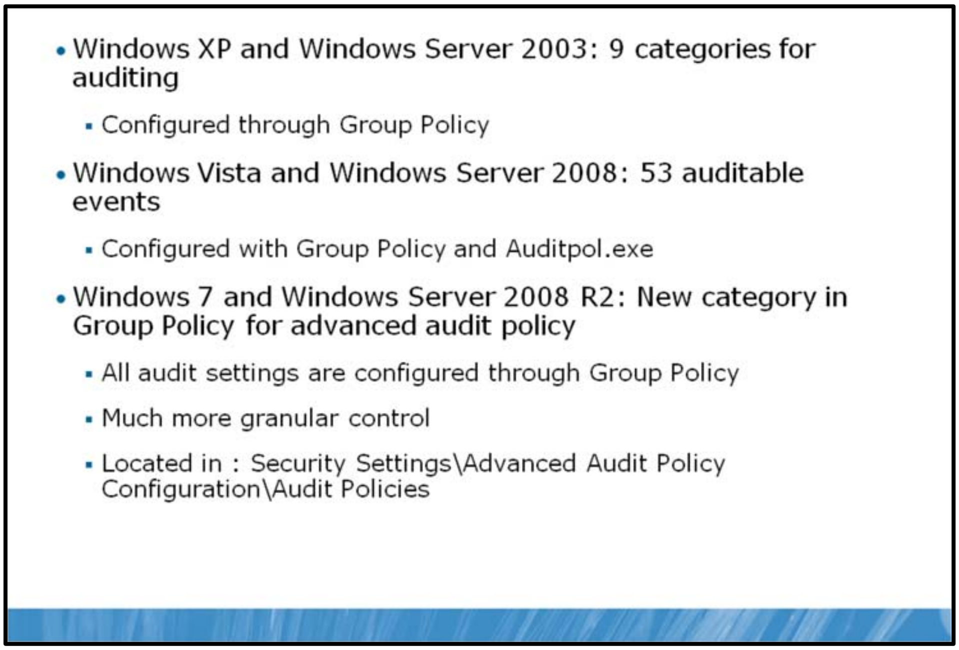


- Event Viewer → Windows Logs → Security
- Each event shows
  - Success/Failure
  - Time
  - Object accessed
  - Identity of user who generated the event
  - Task category

After you enable the desired audit policy setting and specify the access you want to audit by using object SACLs, the system begins to log access according to audit entries. You can view the resulting events in the Security Log of the server. Open the Event Viewer console from Administrative Tools. Expand Windows Logs, and select Security Log.

When Directory Service Changes auditing is enabled and auditing entries are configured in the SACL of directory service objects, events are logged to the Security Log that clearly indicate the attribute that was changed and the change made. In most cases, event log entries will show the previous and current value of the changed attribute.

Advanced Audit Policies



- Windows XP and Windows Server 2003: 9 categories for auditing
  - Configured through Group Policy
- Windows Vista and Windows Server 2008: 53 auditable events
  - Configured with Group Policy and Auditpol.exe
- Windows 7 and Windows Server 2008 R2: New category in Group Policy for advanced audit policy
  - All audit settings are configured through Group Policy
  - Much more granular control
  - Located in : Security Settings\Advanced Audit Policy Configuration\Audit Policies

Note The content in this topic is specific to Windows Server 2008 R2.

In the previous versions of Windows, such as Windows XP® and Windows Server® 2003, nine categories for auditing existed. Administrators could configure each category to perform auditing and monitor successful, failed, or both successful and failed attempts for specific tasks and events. These events are fairly broad in scope and can be triggered by a variety of similar actions; some of which can generate a large number of event log entries. This type of auditing was configured by using Group Policy.

In Windows Vista® and Windows Server 2008, the number of auditable events is expanded from nine to 53, which enables an administrator to be more selective in the number and types of events to audit. However, unlike the nine basic Windows XP events, these new audit events are not integrated with Group Policy and can only be deployed by using logon scripts generated with the Auditpol.exe command-line tool. This was somewhat inconvenient because several tools were used to manage auditing.

In Windows Server 2008 R2 and Windows 7, all auditing capabilities have been integrated with Group Policy. This allows administrators to configure, deploy, and manage these settings in the Group Policy Management Console (GPMC) or Local Security Policy snap-in for a local computer, domain, site, or OU. Windows Server 2008 R2 and Windows 7 make it easier for IT professionals to track when precisely defined, significant activities take place on the network.

Audit policy enhancements in Windows Server 2008 R2 and Windows 7 allow administrators to connect business rules and audit policies. Using these new policies, you can easily configure auditing that will comply with company policy. These new policies for auditing now have a specific node in the Security settings part of Group Policy object—they are located in Security Settings\Advanced Audit Policy Configuration\Audit Policies. Within this node, there are 10 categories for auditing with several options within each category. At the same time, the legacy audit policy node still exists.

Basic Audit policies vs. Advanced Audit Policies

The basic security audit policy settings (located in Security Settings\Local Policies\Audit Policy) and the advanced security audit policy settings (located in Security Settings\Advanced Audit Policy Configuration\Audit Policies) appear to overlap, but they are recorded and applied differently.

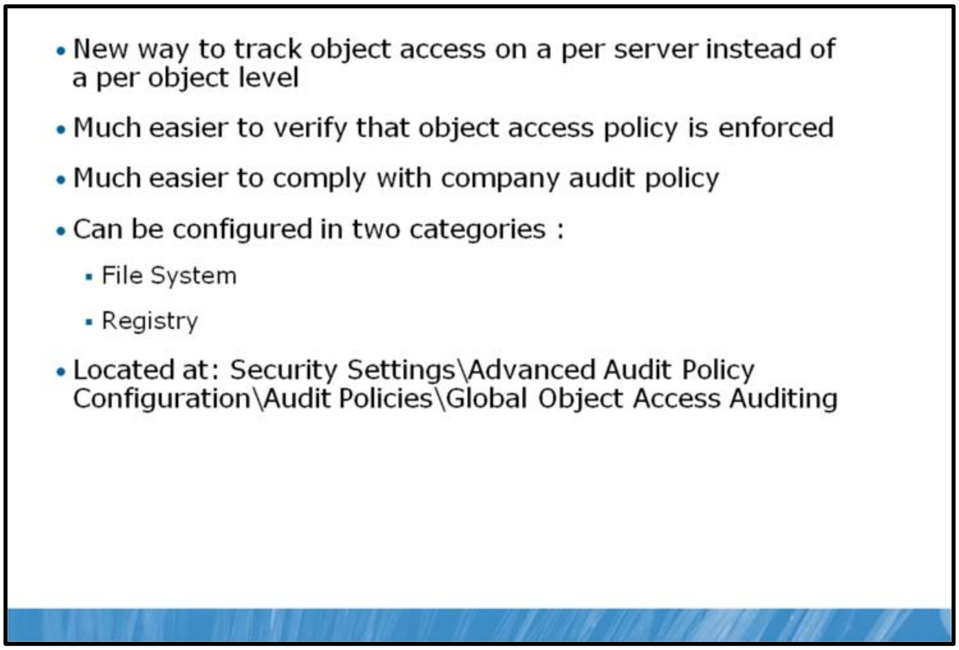
When you apply basic audit policy settings to the local computer by using the Local Security Policy, you are editing the effective audit policy, so changes made to basic audit policy settings appear exactly as configured in Auditpol.exe.

There are several additional differences between the security audit policy settings in these two locations.

A new set of advanced audit policies allow administrators to be more selective in the number and types of events to audit. For example, where a basic audit policy provides a single setting for account logon, advanced audit policy provides four. Enabling the single basic account logon setting would be the equivalent of setting all four advanced account logon settings. In comparison, setting a single advanced audit policy setting does not generate audit events for activities you are not interested in. Additionally, if you enable success auditing for the basic Audit account logon events setting, only success events will be logged for all account logon–related behaviors. In comparison, you can configure success auditing for one advanced account logon setting, failure auditing for a second advanced account logon setting, success and failure auditing for a third advanced account logon setting, or no auditing, depending on the needs of your organization.

The nine basic settings under Security Settings\Local Policies\Audit Policy were introduced in Windows 2000, and therefore are available to all versions of Windows released since then. The advanced audit policy settings were introduced in Windows Vista and Windows Server 2008. The advanced settings can be used only on computers running Windows 7®, Windows Vista, Windows Server 2008, or Windows Server 2008 R2.

Global Object Access Auditing



- New way to track object access on a per server instead of a per object level
- Much easier to verify that object access policy is enforced
- Much easier to comply with company audit policy
- Can be configured in two categories :
  - File System
  - Registry
- Located at: Security Settings\Advanced Audit Policy Configuration\Audit Policies\Global Object Access Auditing

Note The content in this topic is specific to Windows Server 2008 R2.

To enable object access auditing, in previous Windows versions, you had to configure this option in basic audit policies (in GPOs), and also turn on auditing for a specific security principal on SACL of object which you want to audit. This approach sometime was not so easy to adjust with company policies such as "Log all administrative write activity on servers containing Finance information," because you cannot turn on object access audit logging on server level but only on object level.

The new audit category in Windows Server 2008 R2 allows administrators to manage object access auditing in a much wider scope.

With Global Object Access Auditing, administrators can define computer SACLs per object type for either the file system or registry. The specified SACL is then automatically applied to every object of that type.

A global object access audit policy can be used to enforce the object access audit policy for a computer, file share, or registry without configuring and propagating conventional SACLs. Configuring and propagating SACLs is a more complex administrative task, and it is difficult to verify, particularly if you need to verify to an auditor that security policy is being enforced.

Auditors will be able to prove that every resource in the system is protected by an audit policy by just viewing the contents of the Global Object Access Auditing policy setting.

Resource SACLs are also useful for diagnostic scenarios. For example, setting a Global Object Access Auditing policy to log all activity for a specific user and enabling the Access Failures audit policies in a resource (file system, registry) will help administrators quickly identify which object in a system is denying a user access.

Global Object Access Auditing includes the following subcategories: File system and registry.

File System

This security policy setting allows you to configure a global SACL on the file system for an entire computer.

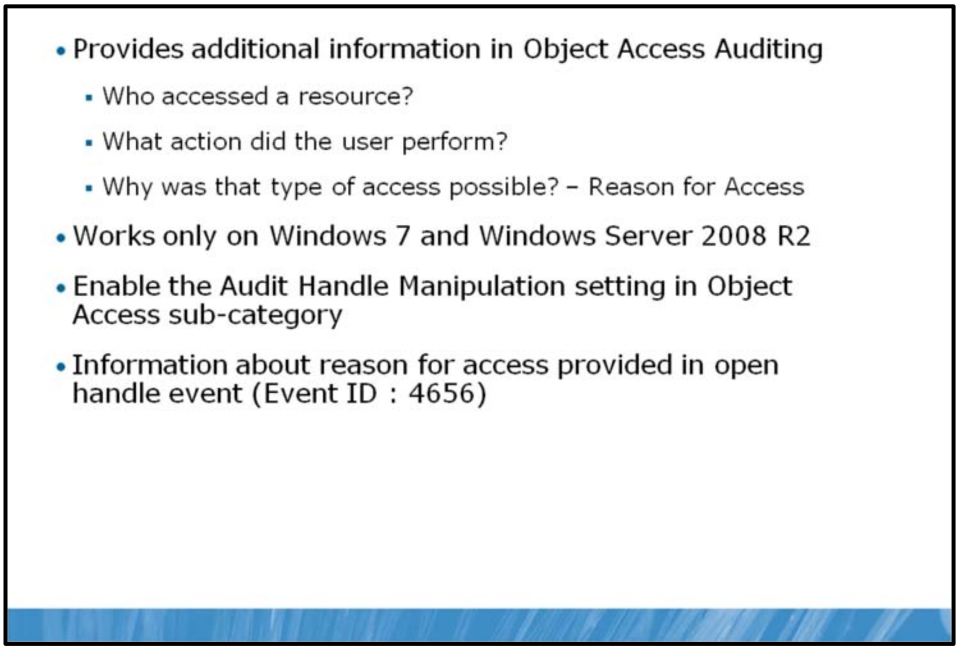
If you select the **Configure security** check box, you can add a user or group to the global SACL.

Registry

This security policy setting allows you to configure an SACL on the registry for a computer. If you select the Configure security check box, you can add a user or group to the global SACL. This policy setting must be used in combination with the Registry security policy setting under Object Access.

Note If both a file or folder SACL and a Global Object Access Auditing policy (or a single registry setting SACL and a Global Object Access Auditing policy) are configured on a computer, the effective SACL is derived from combining the file or folder SACL and the Global Object Access Auditing policy. This means that an audit event is generated if an activity matches either the file or folder SACL or the Global Object Access Auditing policy.

Reason for Access Reporting



- Provides additional information in Object Access Auditing
  - Who accessed a resource?
  - What action did the user perform?
  - Why was that type of access possible? – Reason for Access
- Works only on Windows 7 and Windows Server 2008 R2
- Enable the Audit Handle Manipulation setting in Object Access sub-category
- Information about reason for access provided in open handle event (Event ID : 4656)

Note The content in this topic is specific to Windows Server 2008 R2.

One of the most common auditing needs is to track access to a particular file or folder. There are several events in Windows to audit whenever an object access operation was successful or unsuccessful. The events usually include the user, the object, and the operation, but they lack the reason why the operation was allowed or denied.

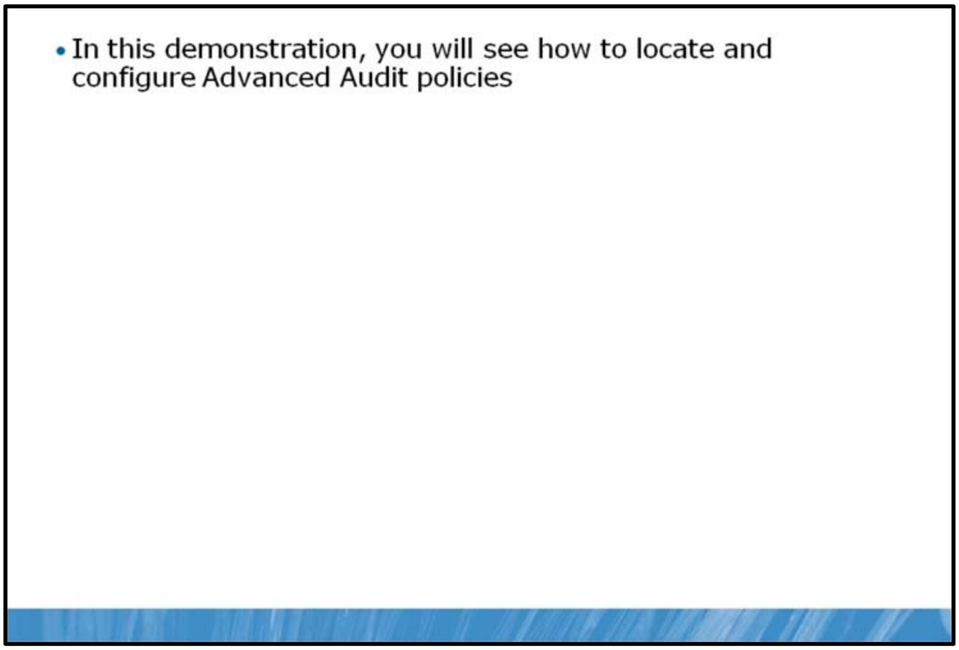
However, it is often not enough to know simply that an object such as a file or a folder was accessed by a user. For example, you might need to identify an activity such as a user writing to a file that he or she should not have had access to. You may also want to know why the user was able to access this resource. Windows Server 2008 R2 and Windows 7 improve this forensics analysis by providing additional information about why someone had access to a specific resource. This feature is called Reason for Access auditing (or reporting).

By enabling Reason for Access auditing, in addition to tracking this type of activity, you will also be able to identify the exact ACE that allowed the undesired access. This can significantly simplify the task of modifying access control settings to prevent similar undesired object access in the future.

In Windows Server 2008 R2 and Windows 7, you can obtain this forensic data by configuring the Audit Handle Manipulation setting along with either the Audit File System or Audit Registry audit settings in Advanced Audit Policy Configuration.

In Windows 7 and Windows Server 2008 R2, the reason why someone has been granted or denied access is added to the open handle event. This makes it possible for administrators to understand why someone was able to open a file, folder, or file share for a specific access.

Demonstration: Advanced Audit Policies



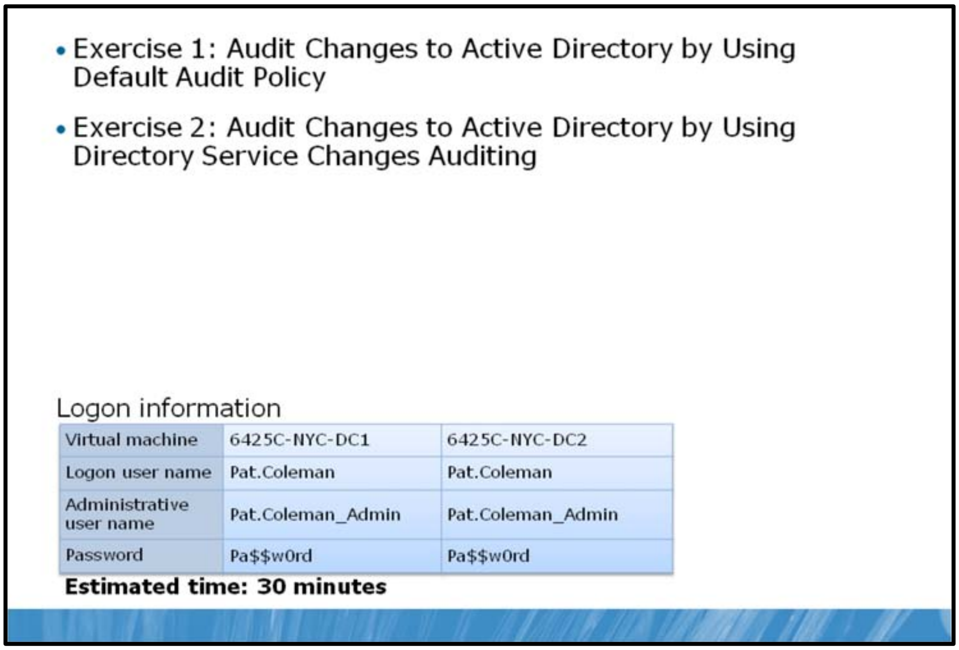
- In this demonstration, you will see how to locate and configure Advanced Audit policies

In this demonstration, you will see how to locate and configure Advanced Audit policies

Demonstration Steps

- Start **Group Policy Management Console**.
- Edit the Default Domain Policy GPO.
- Browse to **Advanced Audit Policy Configuration**.
- Browse to **subcategories**.
- Configure that Advanced Audit Policy Configuration settings are not overwritten.

Lab B: Audit Active Directory Changes



- Exercise 1: Audit Changes to Active Directory by Using Default Audit Policy
- Exercise 2: Audit Changes to Active Directory by Using Directory Service Changes Auditing

Logon information

Virtual machine	6425C-NYC-DC1	6425C-NYC-DC2
Logon user name	Pat.Coleman	Pat.Coleman
Administrative user name	Pat.Coleman_Admin	Pat.Coleman_Admin
Password	Pa$$w0rd	Pa$$w0rd

Estimated time: 30 minutes

Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. On the host computer, click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In Hyper-V Manager, click **6425C-NYC-DC1**, and in the Actions pane, click **Start**.
3. In the Actions pane, click **Connect**. Wait until the virtual machine starts.
4. Log on by using the following credentials:
   - User name: **Pat.Coleman**
   - Password: **Pa$$w0rd**
   - Domain: **Contoso**
5. Repeat **steps 2–4 for 6425C-NYC-DC2.**

Lab Scenario

The administrators at Contoso, Ltd have reported a few times that the membership lists of certain highly privileged groups are inconsistent. The lists included people who should not be members of these groups. One possible reason for the inconsistency could be that the membership list of these groups is changed by following incorrect procedures. The enterprise security team at Contoso, Ltd has asked you to provide detailed reports regarding changes to the membership of security-sensitive groups, including Domain Admins. The reports must show the change that was made, who made the change, and when.

Exercise 1: Audit Changes to Active Directory by Using Default Audit Policy

In this exercise, you will see the Directory Service Access auditing that is enabled by default in Windows Server 2008 and Windows Server 2003.

The main tasks for this exercise are as follows:

1. Confirm that the Domain Admins group is configured to audit changes to its membership.
2. Make a change to the membership of Domain Admins.
3. Examine the events that were generated.

▶ Task 1: Confirm that the Domain Admins group is configured to audit changes to its membership.

1. On NYC-DC2, run **Active Directory Users and Computers** as an administrator, with the user name **Pat.Coleman_Admin** and the password **Pa$$w0rd**.
2. Open the **Audit Settings** properties of the **Domain Admins** group.
3. Locate the entry that specifies the auditing of successful attempts to modify properties of the group such as membership.

   Question: What is the Auditing Entry that achieves this goal?

▶ Task 2: Make a change to the membership of Domain Admins.

1. Add **Stuart Munson** (user logon name **Stuart.Munson**) to the **Domain Admins** group. Be sure to apply your change.
2. Remove **Stuart Munson** from the **Domain Admins** group.
3. Make a note of the time when you made the changes. That will make it easier to locate the audit entries in the event logs.

▶ Task 3: Examine the events that were generated.

1. Run **Event Viewer** as an administrator, with the user name **Pat.Coleman_Admin** and the password **Pa$$w0rd**.
2. Click **Security Log** and locate the events that were generated when you added and removed Stuart Munson.

   Question: What is the Event ID of the event logged when you made your changes? What is the Task Category?

   Question: Examine the information provided on the General tab. Can you identify the following in the event log entry?

   - Who made the change?
   - When the change was made?
   - Which object was changed?

- What type of access was performed?
- Which attribute was changed? How is the changed attribute identified?
- What change was made to that attribute?

Results: In this exercise, you generated and examined Directory Service Access audit entries.

Exercise 2: Audit Changes to Active Directory by Using Directory Service Changes Auditing

In this exercise, you will implement the new Directory Services Changes auditing of Windows Server 2008 to reveal details about changes to the Domain Admins group.

The main tasks for this exercise are as follows:

1. Enable Directory Services Changes auditing.
2. Make a change to the membership of Domain Admins.
3. Examine the events that were generated.

▶ Task 1: Enable Directory Services Changes auditing.

1. On NYC-DC2, run the command prompt as an administrator, with the user name **Pat.Coleman_Admin** and the password **Pa$$w0rd.**
2. Type the following command, and then press Enter.

```
auditpol /set /subcategory:"directory service changes" /success:enable
```

▶ Task 2: Make a change to the membership of Domain Admins.

1. Add **Stuart Munson** (user logon name **Stuart.Munson**) to the **Domain Admins** group. Be sure to apply your change.
2. Remove **Stuart Munson** from the **Domain Admins** group.
3. Make note of the time when you made the changes. That will make it easier to locate the audit entries in the event logs.

▶ Task 3: Examine the events that were generated.

1. Run **Event Viewer** as an administrator, with the user name **Pat.Coleman_Admin** and the password **Pa$$w0rd**.
2. Click **Security Log** and locate the new types of events that were generated when you added and removed Stuart Munson.

   Question: What are the Event IDs of the event logged when you made your changes? What is the Task Category?

   Question: Examine the information provided on the General tab. Can you identify the following in the event log entry?

   - What type of change was made?
   - Who made the change?
   - Which member was added or removed?
   - Which group was affected?
   - When the change was made?

Results: In this exercise, you generated Directory Services Changes auditing entries.

▶ To prepare for the next module

When you finish the lab, revert the virtual machines to their initial state. To do this, complete the following steps:

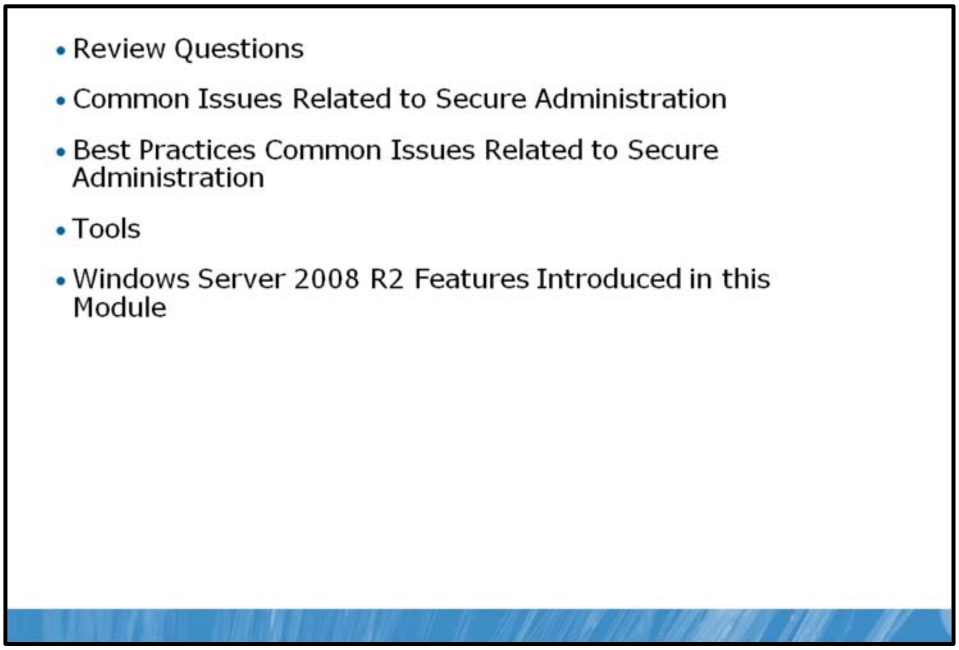
1. On the host computer, start Hyper-V Manager.
2. Right-click **6425C-NYC-DC1** in the **Virtual Machines** list, and then click **Revert**.
3. In the **Revert Virtual Machine** dialog box, click **Revert**.
4. Repeat these steps for **6425C-NYC-DC2.**

Lab Review Questions

Question: What details are captured by Directory Services Changes auditing that are not captured by Directory Service Access auditing?

Question: Which type of administrative activities would you want to audit by using Directory Services Changes auditing?

Module Review and Takeaways



- Review Questions
- Common Issues Related to Secure Administration
- Best Practices Common Issues Related to Secure Administration
- Tools
- Windows Server 2008 R2 Features Introduced in this Module

Review Questions

Question: How does the Active Directory Users and Computers console indicate that you do not have permissions to perform a particular administrative task?

Question: What is the benefit of a two-tiered, role-based management group structure when assigning permissions in Active Directory?

Note Role-based management is a detailed topic. There are other aspects of role-based management such as discipline and auditing that are required to ensure that the members of a group such as AD_User Accounts_Support have the permissions they are supposed to have. You also need to ensure that the members of this group have no other permissions, and that no other users or groups have been delegated the same permissions.

Question: What is the main benefit of using new Advanced Audit Policies?

Common Issues related to Secure Administration

Issue	Troubleshooting tip
There is no un-delegate command or wizard after you finish delegation of control	
Reason for Access auditing is not working	

Best Practices Related to Secure Administration

- Use Delegation of Control Wizard to delegate administrative control instead of placing users in built-in administrative groups.

- Use Advanced Audit Policies for better and more granular audit control.
- Avoid using the block inheritance option when configuring permissions.

Tools

Tool	Used for	Where to find it
Group Policy Management Console	Editing security policy	Administrative Tools
Delegation of Control Wizard	Delegating administrative control over OU	Active Directory Users and Computers
Auditpol	Configuring auditing	Command-line utility

Windows Server 2008 R2 Features Introduced in this Module

Windows Server 2008 R2 feature	Description
Advanced Audit Policies	New settings in Group Policy object for more detailed auditing of various system events
Global Object Access Auditing	Method to audit on server level instead on object level
Reason for access reporting	New feature that allows administrators to see why someone was able to access a resource that is being audited.

Module 10

Improving the Security of Authentication in an AD DS Domain

Contents:

Lesson 1: Configure Password and Lockout Policies 10-3

Lab A: Configure Password and Account Lockout Policies 10-19

Lesson 2: Audit Authentication 10-24

Lab B: Audit Authentication 10-31

Lesson 3: Configure Read-Only Domain Controllers 10-34

Lab C: Configure Read-Only Domain Controllers 10-49